A Study on Big Data Integration with Data Warehouse

T.K.Das[1] and Arati Mohapatro[2]
[1](School of Information Technology & Engineering, VIT University, Vellore ,India)
[2](Department of Computer Science, Bangalore City College, Bangalore, India)

Abstract -*The amount of data in world is exploding. Data is being collected and stored at unprecedented rates. The challenge is not only to store and manage the vast volume of data, but also to analyze and extract meaningful value from it. In the last decade Data Warehousing technology has been evolved for efficiently storing the data from different sources for business intelligence purpose. In the Age of the Big Data, it is important to remodel the existing warehouse system that will help you and your organization make the most of unstructured data with your existing Data Warehouse. As Big Data continues to revolutionize how we use data, this paper addresses how to leverage big data by effectively integrating it to your data warehouse.*

Keywords *- Big Data, Data warehouse, Hadoop*

1. INTRODUCTION

We have data warehouses built using relational technology mainly for operational sources. Big data comes from relatively new types of data sources like social media, public filings, content available in the public domain through agencies or subscriptions, documents and e-mails including both structured and unstructured texts, digital devices and sensors including location-based smart phone, weather and telemetric data .Companies aren't accustomed to collecting information from these sources, nor are they used to dealing with such large volumes of unstructured data. Therefore, much of the information available to enterprises isn't captured or stored for long-term analysis, and opportunities for gaining insight are missed. Because of the huge data volumes, many companies do not keep their big data, and thus do not realize any value out of this. Big Companies that want to truly benefit from big data must also integrate these new types of information with traditional corporate data, and fit the insight they glean into their existing business processes and operations.There are several approaches to collecting, storing, processing, and analyzing big data. The main focus of the paper is on unstructured data analysis. Unstructured data refers to information that either does not have a pre-defined data model or does not fit well into relational tables. Unstructured data is the fastest growing type of data, some example could be imagery, sensors, telemetry, video, documents, log files, and email data files. There are several techniques to address this problem space of unstructured analytics. The techniques share common characteristics of scale-out, elasticity and high availability. MapReduce, in conjunction with the Hadoop Distributed File System (HDFS) and HBase database, as part of the Apache Hadoop project is a modern approach to analyze unstructured data. Hadoop clusters are an effective means of processing massive volumes of data, and can be improved with the right architectural approach. As enterprises adopt the Hadoop framework for unstructured data analytics, a key consideration is to integrate and interface with legacy data warehouse and relational database systems. This paper focuses on the unstructured aspect of big data, features of hadoop, advantage and disadvantage of hadoop. Also it discusses whether hadoop is replacement for data warehouse. This paper reviews non-relational big data approaches (NOSQL) such as distributed/shared-nothing architectures, horizontal scaling, key/value stores, and eventual consistency. This part of the paper differentiates between structured versus unstructured data. The paper describes various building blocks and techniques for Map Reduce and HDFS, HBase and their implementation in an open source Hadoop

2. BIG DATA TECHNOLOGIES

2.1. Hadoop

Hadoop [5] is a free, Java-based programming framework that supports the processing of large data sets in a distributed computing environment. It is part of the Apache project sponsored by the Apache Software Foundation. Hadoop provides a parallel storage and processing framework.it runs MapReduce[4] batch programs in parallel on thousands of nodes. Hadoop is a kind of MAD system [3] meaning that (i) it is capable of

attracting all data sources (M standing for Magnetism), (ii) it is capable of adapting its engines to evolutions that may occur in big data sources (A standing for Agility), (iii) it is capable of supporting depth analytics over big data sources much more beyond the possibilities of traditional SQL-based analysis tools (D standing for Depth)[14]

2.2 Hadoop Distributed Filesystem

The Hadoop Distributed File system (HDFS) [6] is a scalable distributed file system that provides high-throughput access to application data. HDFS is written in the Java programming language. A HDFS cluster operates in a master-slave pattern [16], consisting of a master name node and any number of slave data nodes. The name node is responsible for managing the file system tree, the metadata for all the files and directories stored in the tree, and the locations of all blocks stored on the data nodes. Data nodes are responsible for storing and retrieving blocks when the name node or clients request them.

2.3 MapReduce

MapReduce is programming models on top of HDFS for processing and generating large data sets which was developed as an abstraction of the map and reduce primitives present in many functional languages [4, 7]. The abstraction of parallelization, fault tolerance, data distribution and load balancing allows users to parallelize large computations easily [16]. The map and reduce model works well for Big Data analysis because it is inherently parallel and can easily handle data sets spanning across multiple machines. Each MapReduce program runs in two main phases: the map phase followed by the reduce phase. The programmer simply defines the functions for each phase and Hadoop handles the data aggregation, sorting, and message passing between nodes. There can be multiple map and reduce phases in a single data analysis program with possible dependencies between them. Map Phase. The input to the map phase is the raw data. A map function should prepare the data for input to the reducer by mapping the key to the the value for each “line” of input. The key-value pairs output by the map function are sorted and grouped by key before being sent to the reduce phase. The input to the reduce phase is the output from the map phase, where the value is an iterable list of the values with matching keys. The reduce function should iterate through the list and perform some operation on the data before outputting the final result.

3. EXISTING DATA WAREHOUSE

3.1 Existing Data Warehouse

In the 1990s, Bill Inmon defined a design known as a data warehouse. According to him, data warehouse is a subject oriented, integrated, time variant and non volatile collection of data. Basically data warehouse is designed for storing data from operational sources to get a insight from those data. Most of the data from operational sources comes in structured format. The traditional method of managing structured data includes a relational database and schema to manage the storage and retrieval of the dataset. For managing large datasets in a structured fashion, the primary approaches are data warehouses and data marts. A data warehouse is a relational database system used for storing, analyzing, and reporting functions. The data mart is the layer used to access the data warehouse. A data warehouse focuses on data storage. The data from disparate sources is cleaned, transformed, loaded into a warehouse so that it is made available for data mining and online analytical functions. The data warehouse and marts are SQL (Standard Query Language) based databases systems. The two main approaches to storing data in a data warehouse are the following[15]:

• Dimensional: Transaction data are partitioned into “facts” tables, which are generally transaction data and dimensions tables, which are the reference information that gives context to the facts

. • Normalized: The tables are grouped together by subject areas that reflect data categories, such as data on products, customers, and so on. The

normalized structure divides data into entities, which create several tables in a relational database

3.2 Limitations of current DWH practices

i) Limited capability of handling unstructured data. Structured data has a fixed schema so that to fit it a relational model can be designed [9] .On the other hand unstructured and semi structured data do not have a fixed schema. Now-a-days in enterprises these types of data are becoming common .These data also hides prudent information. So the decision making process in enterprise would not be effective as these unstructured data are not being .taken into account while decision making. But traditional data warehouse engine has a limitation in this sphere

iii) Extracting information from big data and matching with existing data.

Big data comes in various formats. Most of the data are unstructured. There is a significant format variation in unstructured text and entities in the database. Querying unstructured data is also not an easy task. Unstructured data could be stored in LOBs, but there is no effective querying inside lob. Though now SQL supports keyword search based on full text, but it is limited to text searching only.From unstructured text, the extracted information are obtained as a sequence of tokens But the database is a form of entity relationship. There is no information in the database about inter entity sequencing.

4. INTEGRATING BIG DATA WITH DATA WAREHOUSE

Most of the organization including millions of customers processing pet bytes of data daily has the following requirement [21]

- Fast data loading
- Fast query processing
- Highly efficient storage utilization
- Strong adaptivity to highly dynamic workload patterns

4.1 Hadoop and Data Warehouse

Hadoop can be reasonably considered as the evolution of next-generation Data Warehousing systems, with particular regards to the ETL phase of such systems. MapReduce is the core of Hadoop. MapReduce is a programming model with the associated computational framework that is inspired to the primitives Map and Reduce of functional languages. Some of the key points of the Hadoop based technology [17]:

- The pure-store is a write-once, read-many solution e.g. no updates and no alterations of existing structures / files
- It is a full file-store, without any referential integrity .It helps in getting fast performance
- Partitioning is wonderful, but the "columns" that the data is partitioned by, do not exist in the data set anymore – they are put in as part of the directory tree structure.
- Partitioning is truly file splitting in to separate physical directories (sometimes separate physical machines)
- Hadoop is a load and go file management system – meaning you copy raw files in to the Hadoop platform; there is no such thing as "ETL" to getting files into Hadoop. They are copied in, and then the transformation rules must be written in code.
- ELT approach as transformation is done after loading. It equal to "file copy (EL), followed by Insert into new file from old file combined with Transformation rules"
- Hadoop is not an Extract-Transform-Load (ETL) tool. It is a platform that supports running ETL

Some Benefits of this approach include [17,18]:

- Rapid loading by means of file copy
- Readily applying transformation as there is no referential integrity, it is easy to aggregate data
- Distributed computing with MPP / Shared Nothing algorithms
- Automated compression .In some Hadoop implementations file compression is built in, reducing the amount of storage necessary to accommodate the data
- Schema Less (to some extent) storage. You still have to define the columns, and in some cases the base data types for the file in order for the code to "map" to the elements. On the other hand, there are some types of files that just work natively (already mapped), like XML. And still other documents (like TEXT files) that simply work based on internal searching for tags. Schema-less is both a benefit and a drawback.

- No SQL – There is no limitations what standard SQL needs.
- It is not required to normalize data sets so as to avoid joins across node sets.

4.2. Architecture

Hadoop and MapReduce solutions are being deployed in enterprises together with data warehouses. Fig.1 shows how data is shared between different processing and the diversity of data experts who can access both

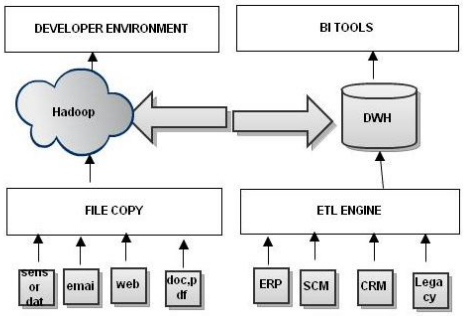

Fig. 1. *Interface between Hadoop and data warehouse*

Complex Hadoop jobs can use the data warehouse as a data source, simultaneously leveraging the massively parallel capabilities of two systems [15]. Any MapReduce program can issue SQL statements to the data warehouse. In one context, a MapReduce program is just another program and the data warehouse is just another database

4.3. Proposed Solution

The big data (mostly unstructured format) are loaded as file to HDFS. The input data are taken as write once fashion. Then it is processed by MapReduce in two phases – Map phase and Reduce phase. The results of processing are written to HDFS. There are two types of HDFS nodes: Data node which stores the data blocks of the files and Name node which consist of metadata.[2]

The following section provides an introduction to MapReduce steps while processing a job [2]

- Input step: Loads the data into HDFS by splitting the data into blocks and also distributing to data nodes of the cluster. The blocks are replicated for availability in case of failures. The Name node keeps track of blocks and the data nodes.
- Job Step: Submits the MapReduce job and its details to the JobTracker.
- Job Init Step: The Job Tracker interacts with TaskTracker on each data node to schedule MapReduce tasks.
- Map step: Mapper process the data blocks and generates a list of key value pairs.
- Sort step: Mapper sorts the list of key value pairs.
- Shuffle step: Transfers the mapped output to the reducers in a sorted fashion.
- Reduce step: Reducers merge the list of key value pairs to generate the final result.
- Finally, the results are stored in HDFS

The results are then retrieved from HDFS by user by writing a java program.

Next the data is loaded in big dimension table of data warehouse by usual data loading mechanism.

As data does not have fixed schema enters into data warehouse, to handle frequently schema changes in the huge fact table, versioning can be done on the basis of schema changes in the fact table e.g. each row in fact table contains a partaker id for a schema version. From that we can know which columns are available for that particular version so that adding or removing columns are easier.

5. RELATED WORK

There has been a lot of recent work on petabyte scale data processing systems. Facbook developed Hive[8] an open source data warehousing solution built on top of Hadoop. It is based on familiar concepts of tables, columns and partitions, providing a high-level query tool for accessing data from their existing Hadoop warehouses [10]. The result is a data warehouse layer built on top of Hadoop that allows for querying and managing structured data using a familiar SQL-like query language, HiveQL. It includes a system catalog - Metastore.In facebook Hive warehouse contains tens of thousands of tables and over 700Tb of data use by reporting and analysis.

Cheetah[20] is designed specifically for an online advertising application to allow various simplifications and custom optimizations. Cheetah data warehouse system built on top of the MapReduce technology. In this virtual view is defined on top of the common data warehouse star/snowflake schema. Ad-hoc MapReduce programs can access the raw data by exploiting this virtual view interface.
Scope[14] is an SQL-like language on top of Microoft's proprietary Cosmos map/reduce and distributed file system. Pig[19] allows users to write declarative scripts to process data.

Informatica 9.5 provides multiple big data capabilities that will allow enterprises to take advantage of multiple big data attribute. Informatica [22] new release adds new support for Hadoop, natural language processing, social networking data which Informatica supports with its Social Master Data Management (MDM) offering. The unstructured data transformation has services like parser, serializer, mapper, transformer and streamer that transforms data

6. CONCLUSION

In this paper we outlined big data technologies and limitation of traditional SQL based data warehouse system .To meet any big data challenges in terms of fast processing and accommodating highly varying big data, the data warehouse must integrate with hadoop engine to avail its Map Reduce parallel processing power. We have explained a common interface between so that a data warehouse can be built on top of hadoop engine. More work can be done to integrate the hadoop based data warehouse with commercial BI tools like Microstrategy , Cognos.

REFERENCES

[1] Abouzeid, A., Bajda-Pawlikowski, K., Abadi, D.J.. Rasin, A., and Silberschatz, A. HadoopDB: An Architectural Hybrid of MapReduce and DBMS Technologies for Analytical Workloads. PVLDB 2(1), 2009..
[2] Bakshi Kapil, Considerations for Big Data –Architecture and Approach, IEEE,2012
[3] Cohen, J., Dolan, B., Dunlap, M., Hellerstein, J.M., and Welton, C. MAD Skills: New Analysis Practices for Big Data. PVLDB 2(2), 2009.
[4] J. Dean and S. Ghemawat. MapReduce: simplified data processing on large clusters.Communications of the ACM, 51(1):107–113, 2008.
[5] Hadoop. http://hadoop.apache.org.
[6] Hadoop distributed file system (hdfs). http://hadoop.apache.org/hdfs.
[7] Hadoop MapReduce. http://hadoop.apache.org/mapreduce.
[8] Hive. http://hive.apache.org.
[9] Liu Z.H., Krishnamurthy.V. ,Towards Business Intelligence over Unified Structured and Unstructured Data using XML, edited volume "Business Intelligence-Solution for Business Development", Intech Publisher,2011
[10] Thusoo, A. Sarma, J.S., Jain, N., Shao, Z., Chakka, P. Zhang, N.,Antony, S., Liu, H., and Murthy, R. Hive – A Petabyte Scale Data Warehouse Using Hadoop. Proc. of ICDE, 2010.
[11] R. Chaiken, et. al. Scope: -Easy and Efficient Parallel Processing of Massive Data Sets. In Proc. of VLDB, 2008..
[12] A. Pavlo, E. Paulson, A. Rasin, D.J. Abadi, D.J. DeWitt, S. Madden, and M. Stonebraker.
A comparison of approaches to large-scale data analysis. In Proceedings of the 35th SIGMOD international conference on Management of data, SIGMOD '09, pages 165–178. ACM, 2009.
[13] M. Stonebraker, D. Abadi, D.J. DeWitt, S. Madden, E. Paulson, A. Pavlo, and A. Rasin. MapReduce and parallel DBMSs: friends or foes? Communications of the ACM, 53(1):64–71, 2010.
[14] Cuzzocrea.A , Song .Y , Davis Karen C : Analytics over large scale Multidimensional Data:The Big Data Revolution, Communications of ACM,2011
[15] Awadallah Amar,Graham.Dan,Hadoop and the Data Warehouse- When to use which , Cloudera Inc and Teradata Corporation,2011
[16] Hollingsworth.A, Graham.D, Hadoop and Hive as scalable alternatives to RDBMS –A Case Study,Boise State University Scholarworks,2012
[17] http://infolab.stanford.edu/~ragho/hive-icde2010.pdf
[18] www.datavault.com
[19] Hadoop Pig. Available at http://hadoop.apache.org/pig
[20] Chen Songting. "Cheetah – Ahigh performance custom Datawarehouse on top of MapReduce" Proceedings of VLDB ,Vol 3,No . 2, 2010
[21] Yongqiang He et all " RCFile: A Fast and Space-efficient Data Placement
Structure in MapReduce-based Warehouse Systems", ICDE 2011
[22] www.informatica.com